AD-A071 137 MICHIGAN UNIV ANN ARBOR DEPT OF ELECTRICAL AND COMPU--ETC F/G 20/4
THE PERFORMANCE OF A VECTORIZED 3-D NAVIER-STOKES CODE ON THE C--ETC(U)
JUL 79 J S SHANG, P G BUNING, W L HANKEY AFOSR-75-2812
UNCLASSIFIED AFOSR-TR-79-0790 NL

| OF |
AD
A071137

END
DATE
FILMED
8-79
DDC

THE PERFORMANCE OF A VECTORIZED 3-D NAVIER-STOKES CODE ON THE CRAY-1 COMPUTER

J. S. Shang*, P. G. Buning**, W. L. Hankey*** and M. C. Wirth****
Air Force Flight Dynamics Laboratory
Wright-Patterson AFB, Ohio

Abstract

A three-dimensional, time dependent Navier-Stokes code using MacCormack's explicit scheme has been vectorized for the CRAY-1 computer. *Computations* were performed for a turbulent, transonic, normal shock wave boundary layer interaction in a wind tunnel diffuser. The vectorized three-dimensional Navier-Stokes code on the CRAY-1 computer achieved a speed of 128 times that of the original scalar code processed by a CYBER 74 computer. The vectorized version of the code outperforms the scalar code on the CRAY computer by a factor of 8.13. A comparison between the experimental data and the numerical simulation is also made.

Nomenclature

c	Speed of Sound
Def	Deformation Tensor
e	Specific Internal Energy $c_v T + (u^2 + v^2 + w^2)/2$
$\bar{F}, \bar{G}, \bar{H}$	Vector Fluxes, Equation (15)
i, j, k	Indexes of the Grid Point System
K	Exponent in the Stretched Coordinates y, z
L	Length Scale of Eddy-Viscosity, Equation (6)
L_ξ, L_η, L_ζ	Differencing Operator
M	Mach Number
$\bar{n}$	Outward Normal of Solid Contour
P	Static Pressure
Pr	Prandtl Number
Pr_t	Turbulent Prandtl Number
$\dot{q}$	Rate of Heat Transfer
q	Magnitude of Velocity
R_{ey}	Reynolds Number Based on Running Length $\rho_\infty u_\infty x/\mu_\infty$
T	Static Temperature
t	Time
$\bar{U}$	Dependent Variables in Vector Form $(\rho, \rho u, \rho v, \rho w, \rho e)$
$\bar{u}$	Velocity Vector
u, v, w	Velocity Components in Cartesian Frame
x, y, z	Coordinates in Cartesian Frame
x_L, y_L, z_L	Length Scales of Cartesian Coordinates
δ_{ij}	Kronecker Delta
ε	Eddy Viscosity Coefficient
ξ, η, ζ	Transformed Coordinate System, Equation (14)
μ	Molecular Viscosity Coefficient
ρ	Density
$\bar{\tau}$	Stress Tensor

Subscripts

∞	Property Evaluated at the Freestream Condition
o	Stagnation Condition

Superscripts

—	Denotes Vector
~	Denotes Tensor
n	Denotes Time Level

Introduction

In the past decade, *computational fluid dynamics* has become firmly established as a credible tool for aerodynamics research. Significant results have been obtained for the inviscid-viscous interactions including flow separation and flows over arbitrary aerodynamic configurations for a wide range of Reynolds numbers and Mach numbers[1,2]. Aided by some rather crude and heuristic turbulence models, success has been achieved even for complex turbulent flows[3-8]. In spite of all these convincing demonstrations, the objective of a wide application of computational fluid dynamics in engineering design has yet to be achieved. The basic limitation is in cost effectiveness. A lower cost and systematic methodology needs to be developed. From the viewpoint of computational fluid dynamics, the obstacles involve several key issues such as: the efficiency of computational algorithms, the numerical resolution, the automatic gridpoint generation, the turbulence modeling, and the supporting computing facilities. Substantial progress has been made in these interlocked problem areas, but continuous efforts are still in demand[9].

The present analysis addresses one of the key objectives in obtaining efficient numerical processing. To achieve this objective, two approaches seem obvious; either develop special algorithms designed for a particular category of problems according to the laws of physics or utilize an

* Aerospace Engineer, Flight Mechanics Division, Member AIAA
** Graduate Student University of Michigan, now with Stanford University
*** Senior Scientist, Flight Mechanics Division, Associate Fellow, AIAA
****Computational Physicist, Computer Support Branch

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF TRANSMITTAL TO DDC
This technical report has been reviewed and is approved for public release IAW AFR 190-12 (7b).
Distribution is unlimited.
A. D. BLOSE
Technical Information Officer

improved computer. In the case of special algorithms, a better understanding of the generic structure of the flow field is required. Time-dependent Navier-Stokes equations[3-8], parabolized Navier-Stokes equations[10,11], boundary-layer approximations or inviscid assumptions have been developed to satisfy special needs. Efforts to develop more efficient finite differencing schemes have lead to a group of implicit[12,13] and hybrid procedures[14,15,16]. In general, these attempts have been successful and have achieved an order of magnitude improvement in computing speed. On the other hand, a class of computers designed for scientific computations; the CRAY-1, STAR 100 and ILLIAC IV among others, has become available. The most significant advance in computer hardware related to computational fluid dynamics is the vector processor which permits a vector to be processed at an exceptional speed. This option gives a new perspective; ie, a drastic reduction in computing time.

Several efforts on vectorization of computer codes with varying degrees of success have been reported in the open literature [17,18,19,20]. The potential of increased speed in data processing rate is clear. Basically, the objective of the vectorization of a computer code is to construct a long ordered data string according to the computer structure to be processed by the vector registers and to achieve optimum data flow.

A three-dimensional time dependent Navier-Stokes code using MacCormack's explicit scheme[21] has been vectorized for the CRAY-1 computer. The selection of this particular finite differencing scheme is based on its past ability to perform a large number of successful bench mark runs[2-7], its proven shock-capturing capability, and the inherent simplicity of the basic algorithms. The CRAY-1 computer was chosen because at the present time, among all the available general purpose scientific processors, it provides the highest potential floating point computation rate in both the scalar and the vector mode. Floating point operations per second, FLOPS, may be used as a criterion for the measure of the central process or unit speed. The asymptotic rate for processing long vectors on the CRAY-1 is 160×10^6 floating point operations per second (FLOPS)[22]. The combination of the selected algorithm and the CRAY-1 computer provides a bench mark for future development and a tool for current engineering evaluation.

The problem selected for evaluating the CRAY-1 performance was the experimental investigation of Abbiss[23,24] of a three-dimensional interaction of a normal shock with a turbulent boundary layer in a square wind tunnel diffuser at a Reynolds number of thirty million and Mach number of 1.51. The primary purpose of the paper is to determine the computational speed of the code, although a comparison with experimental data is presented to demonstrate the validity of the solution.

Governing Equations

The time dependent, three dimensional compressible Navier-Stokes equations in mass-averaged variables can be given as

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \bar{u}) = 0 \qquad (1)$$

$$\frac{\partial \rho \bar{u}}{\partial t} + \nabla \cdot (\rho \bar{u}\bar{u} - \bar{\tau}) = 0 \qquad (2)$$

$$\frac{\partial \rho e}{\partial t} + \nabla \cdot (\rho e \bar{u} - \bar{u} \cdot \bar{\tau} + \dot{\bar{q}}) = 0 \qquad (3)$$

The turbulent closure of the present analysis is accomplished through an eddy viscosity model. The effective thermal conductivity is also defined by the turbulent Prandtl number ($Pr_t = 0.9$). An albegraic two-layer model of the Cebeci-Smith type was used where:

$$\varepsilon_i = \rho (k_1 L)^2 \{1 - [\exp(-\rho \left|\frac{\partial q}{\partial n}\right| L^2)^{1/2} / 26]\}^2 \frac{\left|\frac{\partial q}{\partial n}\right|}{\mu} \qquad (4)$$

$$\varepsilon_o = \rho k_2 q_{max} \int_0^{y_{max}} (1 - q/q_{max})\, dy \qquad (5)$$

k_1 is the Von Karman constant (0.40), and k_2 assumes the value of 0.0168. The length scale L is the asymptotic form given by Gessner for a rectangular duct[25]

$$L = 2yz/[y + z + (y^2 + z^2)^{1/2}] \qquad (6)$$

The components of shear stress and heat flux vector thus can be given as

$$\tau_{ij} = (\mu + \varepsilon)(\mathrm{Def}\ \bar{u})_{ij} - [2/3\ (\mu + \varepsilon)(\nabla \cdot \bar{u}) + p]\, \delta_{ij} \qquad (7)$$

$$\dot{q}_i = -C_p \left(\frac{\mu}{Pr} + \frac{\varepsilon}{Pr_t}\right) \frac{\partial T}{\partial x_i} \qquad (8)$$

The equation of state, Sutherland's viscosity law and assigned molecular Prandtl number (0.73) formally close the system of governing equations.

Since the wind tunnel flow field consisted of four symmetrical quadrants, only a single quadrant was computed. The boundaries of the computational domain contain two intersecting wind tunnel walls and two planes of symmetry for which the associated boundary conditions are straight forward (Figure 1). In order to develop upstream conditions equivalent to the experiment a separate computation is initiated with a free stream condition and permitted to develop a three-dimensional boundary layer along the corner region until the boundary layer duplicates the experimental observation ($\delta = 4.0$ cm, $x = 316$ cm)[23]. Then, the computed flow field at this streamwise location is imposed as the upstream condition for the interaction computation. On the wind tunnel walls, the boundary conditions are no-slip for the velocity components and a constant surface temperature. The wind tunnel wall pressure is obtained by satisfying the momentum equation at the solid surface. On the planes of symmetry, the symmetrical boundary conditions are given for all dependent variables. The normal shock wave across the wind tunnel is then specified according to the Rankine-Hugoniot conditions. The far downstream boundary condition is the well known no-change condition. In summary

Initial condition:

$$\bar{U}(0, \xi, \eta, \zeta) = \bar{U}_\infty \tag{9}$$

Upstream condition:

$$\bar{U}(t, 0, \eta, \zeta) = \bar{U}_\infty \tag{10}$$

Downstream:

$$\left.\frac{\partial \bar{U}}{\partial x}\right|_{x \to x_L} = 0 \tag{11}$$

On planes of symmetry:

$$\left.\frac{\partial \bar{U}}{\partial y}\right|_{y = y_L} = 0 \text{ and } \left.\frac{\partial \bar{U}}{\partial z}\right|_{z = z_L} = 0 \tag{12}$$

On wind tunnel wall

$$\left.\begin{aligned} u = v = w &= 0 \\ T_w &= 313.79^0 K \\ \nabla \cdot \bar{\tau} &= 0 \end{aligned}\right\} \text{ at } y, z = 0 \qquad \begin{aligned} &(13a) \\ &(13b) \\ &(13c) \end{aligned}$$

A coordinate system transformation is introduced to improve the numerical resolution in the viscous dominated region,

$$\xi = x/x_L \tag{14a}$$

$$\eta = 1/k \ln[1 + (e^k - 1)\, y/y_L] \tag{14b}$$

$$\zeta = 1/k \ln[1 + (e^k - 1)\, z/z_L] \tag{14c}$$

The governing equations in the transformed space are of the following form:

$$\frac{\partial \bar{U}}{\partial t} + \xi_x \frac{\partial \bar{F}}{\partial \xi} + \sum_i \eta_{x_i} \frac{\partial \bar{G}}{\partial \eta} + \sum_i \zeta_{x_i} \frac{\partial \bar{H}}{\partial \zeta} = 0$$

$$i = 1, 2, 3 \tag{15}$$

where ξ_x, η_y and ζ_z are the metrics of the coordinate transformation. The definition of the conventional flux vectors F, G, and H can be found in Ref. 7.

CRAY-1 Architecture

In order to exploit the full capacity of the CRAY-1 computer, an understanding of the architecture is necessary. A detailed description of the central processing unit, memory section, and information of data flow can be found in Ref. 26. The control of data flow between the parallel functional units and hierarchically organized memories is of fundamental importance. The CRAY-1 memory section normally consists of 16 banks of bi-polar 1024 - bit LSI memory. The memory size can be as large as 1,048,576 words. Each word is 72 bits long and contains 64 data bits and 8 check bits. The memory cycle time is four clock periods (40 n sec). The access time (the time required to fetch an operand from memory to a scalar register) is 11 clock periods (1325 n sec). The maximum transfer rate for the intermediate register, intermediate scalar registers and vector registers is one word per clock period (12.5 n sec). The address and operand data are transmitted along a single path between main and block register memory every clock period. The judicious usage of the relatively small block register memory essentially determines the effectiveness of the vector registers and allows concurrent operations (chaining) within one computer clock period. A major portion of the present effort was to organize the data storage and to reduce the total number of fetch/store data operations in the sweeping sequences.

The effect of I/O on program speed is minimized by overlapping data transfer with computations. CRAY-1 I/O section has twelve input and twelve output channels so arranged that four simultaneous I/O operations can be performed; each at a rate of one word every four clock period if there is no conflict. For each grid point per sweep 18 words of I/O (five dependent variables, seven metrics, one eddy viscosity coefficient and the five updated dependent variables) must be executed. The buffered I/O and multiple independent I/O channels permit the concurrent I/O transfer and arithmetic operations. Based on our estimate, a 64 x 47 x 47 grid-point system could have been solved on the one-million word memory unit of the CRAY-1 using the above procedure.

The vector processor generates results at rates greatly exceeding the rates of conventional scalar processing by performing operations on sets of ordered data. The CRAY-1 has eight 64-element vector registers. All operands processed by the CRAY-1 are held in registers prior to and after being processed by the functional units. In general, the sequence of operations is to load one or more vector registers from memory and pass them to functional units. A result may be received by a vector register and re-entered as an operand to another vector computation in the same clock period. The chaining of two or more vector operations allows the CRAY-1 to produce more than one result per clock period. Chain operation is automatically detected by the CRAY-1, but certain reordering of the code segments may determine the chain operation. A long vector which exceeds 64 elements, is processed as one or more 64-element segments and a possible remainder of less than 64 elements. No rigid requirement is imposed to construct the vector loop.

From a code developer's viewpoint, the construction of a vectorized code hinges on how to avoid long or overly complicated loops, the non unit increment of subscripts, nonlinear indexing, and logical statement inside of a "do loop". A major improvement usually results if the scalar temporary variables which are encountered most frequently in the repetitious finite differencing scheme are replaced with the vector temporary variables. The recursive loop of which the output is propagated back into the input should be eliminated if the vector element is less than 64. The few simple rules suggested by Higbie were found to be extremely useful[27].

Numerical Procedure and Data Structure

The basic numerical method is the time-split or factorized scheme originated by MacCormack. The finite difference formulation in terms of the difference operator can be expressed as

$$\bar{U}^{n+2} = \sum_n L_\zeta \left(\frac{\Delta t}{2n}\right) \sum_m L_\eta \left(\frac{\Delta t}{2m}\right) L_\xi \, (\Delta t)$$

$$\sum_m L_\eta \left(\frac{\Delta t}{2m}\right) \sum_n L_\zeta \left(\frac{\Delta t}{2n}\right) \qquad (16)$$

Each difference operator contains a predictor and corrector. During a specific numerical sweep, the flux vectors are approximated by a central, forward, and backward differencing scheme in such a fashion that after a complete cycle of the predictor and corrector operations all the derivatives are effectively approximated by a central differencing scheme. A graphic representation of these operations is given by Figure 2. For a numerically simulated three-dimensional flow field, the storage in two time-level memory of all dependent variables, metrics of the Jacobian and the eddy viscosity coefficient requires core storage usually exceeding the capacity of currently available computers. For vector processing an even more stringent requirement on data organization is required.

In the scalar version of the code, the core limitation was remedied by organizing the data according to its streamwise location into planar storage (pages). Only the data in process were brought into the central memory core, while the remainder were retained on a random access disk file. Three pages of predictor level dependent variables, four pages of corrector level dependent variables, and two pages of transformation derivatives were required to process a planar sweep. The additional page for the corrector sweep is due to the numerical smoothing scheme originated by MacCormack[28].

When investigating flows with strong shock waves, it is necessary to employ numerical damping in a shock-capturing scheme. Fourth-order pressure damping was utilized which generates an artificial viscosity-like term.

$$\Delta t \Delta \xi_i^3 \frac{\partial}{\partial \xi_i} \left[\frac{|u_i| + c}{4p} \frac{\partial^2 p}{\partial \xi_i^2} \right] \frac{\partial \bar{U}}{\partial \xi_i} \qquad i = 1, 2, 3$$

The approximation of second order central differencing for the corrector step required additional grid point information beyond the immediately adjacent planes. The damping terms, however, are effective only in the presence of shock waves where the numerical resolution is degraded.

For the vector code, the organization of data must satisfy dual constraints. The dependent variables must be reduced to a minimum to satisfy an overall memory size restriction and equally important, the partition of the immediately accessible data from the rest which may reside on disk or other mass storage devices must be optimized. The input/output traffic can significantly degrade the computer performance because the data flow to and from the disk or the mass storage requiring memory access is slower than the transfer from memory to the registers. The basic program structure is designed for a flow field characterized by a dominant flow direction, thus planar partioning was chosen over the block partioning. The relative merits have been discussed by Buning[17], and will not be repeated here.

From the symmetric differencing operator sequence of predictor and corrector steps, one detects that the dependent variables in the predictor level can be completely eliminated by retaining only the three cyclic pages currently in use (Figure 3). For a flow field requiring a large amount of data storage, this reduction in memory requirement is substantial. Meanwhile, the paging process is reduced from two sweeps to one. The predictor and corrector sequence is performed within one sweep by overlapping the corrector operation during one fractional time step.

Once the planar or page storage is adopted, the vector length can be determined. Calculations can be performed over the complete page ($\eta - \zeta$ plane) thus yielding a vector length equal to the product of gridpoints in the η and ζ coordinates. However, this particular arrangement may preclude a future requirement for a greater number of grid points to be retained in memory for vector temporary i.e. immediately accessible vector storage. In addition, the CRAY-1 vector registers (and therefore vector operations) are limited to 64 elements, hence extremely long vector elements would not enhance the vectorization. Therefore, separate vectors are constructed for η and ζ directions, yielding vector lengths approximately equal to the number of grid points in each direction. In order to keep all solutions in the same page ($\eta - \zeta$ plane), the streamwise sweep (ξ sweep) is vectorized in the ζ direction.

For the present problem, the computational domain with the dimension of 356.3 cm x 45.5 cm x 45.5 cm is partitioned into two streamwise sections of 64 pages each. Every page contains 33 x 33 grid points in η and ζ coordinates respectively. The problem is solved in two steps. The first computational section generates a three-dimensional boundary layer over a corner which becomes the in-flow boundary condition for the following shock-boundary layer interaction domaon. Both contain 64 x 33 x 33 grid points, but a finer streamwise mesh spacing $\Delta x = 1.27$ cm was used for the interaction zone to gain a finer numerical resolution of the shock-boundary layer interaction. The ratio between the fine and coarse streamwise grid spacing is 0.3063 yielding a mesh size in the interaction zone of 0.3958 of the local boundary-layer thickness (4.0cm)[23]. The cross flow plane grid-point distribution, however, remains identical between the two overlapping segments. The memory requirement for each is about 0.545 million words.

The numerical solution is considered at its steady state asymptote when the maximum difference between two consecutive time levels of the static pressure in the strong interacting zone is less than 0.2 percent. In the leading computational domain the convergence criterion is established similarly but is based on the velocity profiles instead of pressure.

Timing Results

A portion of the present effort is aimed at making internal comparisons of the relative times for various types of functional unit processing and memory loading (I/O) for the vectorized code. A knowledge of relative time expenditure information is important to provide some insight into the

program execution rate. Although this type of data is code dependent, the present example is deemed typical of a large class of Navier-Stokes solvers. The timing information is measured by vector operation counts[17] and shown in Figure 4. It is obvious that the relative usage of the memory path and functional units is dominated by memory loadings (34.6%) and floating point multiplication (33.3%). Within the functional units, the relative usage of the floating point addition and multiplication has the ratio of two to three. The relative usage of the reciprocal approximation is extremely rare, i.e. less than 2%. Access to a limited number of vector registers is required for temporary storage of intermediate results in vector operations. When these are not available, additional memory loadings result. In spite of the high percentage of memory loading, a portion of the vectorized Fortran code has achieved an execution rate of 42.9 MFLOPS[17]. Further improvements still can be made either in Fortran or assembly language versions of the present code. However, we feel an overall execution rate greater than 60 MFLOPS on this size problem is unlikely.

The timing comparison between the scalar version and vectorized version of the same numerical algorithm is very important for the projection of future developments in computational fluid dynamics. Equally important, a comparison between several high speed processors and the CRAY-1 on the three-dimensional aerodynamics simulation program is highly desirable in the evaluation of the vector performances. A basic dilemma exists for the comparative investigation; namely in the process of vectorization significant changes were made either on the amount of computation performed or on the number of subroutine calls made. The final vectorized program usually bears little resemblemce to the original scalar code[7,17]. Substantial improvement in performance of the vectorized code on a scalar machine has also been reported. However, this improvement in performance can be considered as a contribution due to the vectorization process.

In order to perform the comparative study, a criterion must be established. The ultimate evaluation of data processing rate is the computing time. The completely duplicated computations for an identical fluid mechanics problem are usually prohibited by the incore memory and the indexing limitations for various processors. Therefore, one has to accept the rate of data processing as the criterion. The rate of data processing is commonly defined as

$$\text{RDP} = \text{CPU Time}/(\text{Total Number of Grid Points} \times \text{Total Number of Iterations})$$

The particular rate of data processing is most suitable for numerical programs with similar algorithms and convergence rate. If the ratio between field grid points and boundary points can be maintained between two programs then the comparison is particularly meaningful.

In Table 1, the comparison of timing results between the scalar code and vectorized code on the CRAY-1 is presented.

Table 1
The Comparison of Scalar and Vector Processing on CRAY-1

VERSION OF CODE	RDP (Sec/Pts, ITERATIONS)
Scalar	4.761×10^{-4}
Vector	5.861×10^{-5}

The vectorized program outperforms the original scalar code by a factor of 8.13. In Table 2, the timing results of the scalar code and vectorized code performance for four different computers are given.

Table 2
Comparative Timing Results

COMPUTER	VERSION OF CODE	RDP	$(\text{RDP})_{\text{CYBER 74}}/\text{RDP}$
CYBER 74	Scalar	7.48×10^{-3}	1.0
CDC 7600	Scalar	1.45×10^{-4}	5.2
CRAY-1	Scalar	4.76×10^{-4}	15.7
CRAY-1	Vector	5.86×10^{-5}	127.7
STAR 100	Vector (64bit)	1.50×10^{-4}	49.9
STAR 100	Vector (32bit)	6.00×10^{-4}	124.7

A brief description of each running condition for which the timing results were obtained may help with the interpretion of the data[11]. The computations conducted on CYBER 74 and CDC 7600 with a grid point system of (17 x 33 x 33) were performed in the early phase of the present task[7]. On the CYBER 74 computer the data storage problem was overcome by a data manager subroutine in conjunction with a random access disk file. The computation carried out on CDC 7600 used large core memory for all the dependent variables. The I/O requirement is substantial, particularly for the computation performed on the CYBER 74.

The comparison of timing results between the CRAY-1 and STAR 100 needs special attention. The vector code developed by R. Smith at NASA Langley Research Center[16] contains the full complement of the Jacobian of the coordinate transformation (9 metrics), thereby permitting more complex configurations to be simulated. The present code (7 metrics); however, is designed for a flow field containing a definite orientation bias for aerodynamical engineering application. The aforementioned requirement in formulation definitely will lead to more computations to be performed than for the present simplification. This difference in code definitely contributes to the rather significantly lower data processing rate than the present result. For the explicit numerical scheme, the numerical results either by the 64 bit or 32 bit arithmetic are nearly identical[16]. Therefore, the accurate assessment of the rate of data processing between the CRAY-1 and STAR 100 should be comparable

for aerodynamics engineering simulations.

Comparisons with Experimental Data

Two experimental investigations[23,24] on the interaction of a normal-shock wave with a turbulent boundary layer were used to compare with the present vectorized CRAY-1 computations. The data of Seddon[24] was collected for the arrangement of a suspended flat plate and a shock generator. A substantial amount of the measurements within the boundary layer were recorded by a series of pressure probes. The velocity flow field data of Abbiss et al[23], however, was obtained by means of a laser anemometer. The test arrangement is simply a wind tunnel in which a normal shock wave is held across the whole test section by an adjustable sonic throat positioned far downstream. This particular configuration is simulated by the present analysis. Unfortunately, little flow field information within the boundary layer domain was available. Therefore, for the purpose of comparison, the combined usage of the two sets of data becomes necessary.

For the normal-shock wave and turbulent boundary layer interactions, the only significant length scale is the turbulent boundary-layer thickness at the initial interception of the shock wave. The streamwise coordinate in the present effort is therefore presented in dimensionless units based upon this boundary layer thickness.

In Figure 5, several velocity profiles across the wind tunnel at a Reynolds number of 3.0×10^7 are presented. This location represents the flow field condition at the end of the leading segment of the computational domain which is also the upstream condition for the following interaction zone. An orderly structure in these velocity distributions can be observed reflecting the symmetrical development of the viscous layer over the corner. In the plane normal to the axis of the wind tunnel and at a distance greater than the boundary layer thickness (4 cm) away from the adjacent wall, the two-dimensional boundary layer structure is clearly exhibited. The present results agree reasonably well with the data of Seddon[24]. The data, however, were collected at a Reynolds number one decade lower than the present condition and at a slightly different Mach number (1.47 v.s. 1.51). At the range of Reynolds numbers considered, the Reynolds number dependence should be scaled out by the boundary layer thickness. An independent boundary-layer calculation using the exact simulated condition was performed that verified this contention. It was found that the difference in magnitude of velocity is a few percent. The present result underpredicts the measured boundary layer thickness[23] by about eight percent.

A direct comparison of several velocity distributions between the data of Abbiss et al[23] and the present calculation is presented in Figure 6 for the interaction region. The data are displayed for fixed x/δ and y coordinates away from the corner domain. The coordinate x is taken in the streamwise direction along the tunnel floor and y normal to the floor. Excellent agreement between the data and calculation is observed for the regions either deeply imbedded within the boundary layer or completely contained in the inviscid domain. The maximum discrepancy between data and calculation is in the lambda wave structure. One of the factors contributing to the disparity is that the present variable mesh distribution does not match perfectly with the experimental data collecting location, thereby requiring an interpolation. Nevertheless, in general, the agreement between data and calculation is very good. The maximum disparity between data and calculations is about 10 percent.

In Figure 7 the Mach number contour is presented in an attempt to compare with the flow field structure given by Abbiss et al[23] in Figure 8. The bifurcation of the normal shock wave is clearly indicated. The strong viscous-inviscid zone extends about four undisturbed boundary layer thicknesses upstream of the normal shock wave. As the boundary layer thickens through the continuous compression of the lambda shock wave system, the local boundary layer exceeds twice the value of the initial boundary layer thickness. One can detect the outward displacement of the Mach contour either from Figure 7 or Figure 8. The calculation nearly duplicates all of the primary features of the experimental observation. However, a difference can be discerned in the dimension of the embedded supersonic zone between the experimental observation and calculation. The local supersonic zone emanates from the expansion due to the total pressure difference between the normal shock and the lambda shock structure and the rapid change in the displacement surface. A few percent disparity in predicting the magnitude of velocity lead to the distinguishable discrepancy in the definition of the embedded supersonic zone. A similar observation may be made for the work of Shea[29] in his investigation of the two-dimensional normal-shock wave and turbulent boundary layer interaction.

In Figure 9, the velocity distribution parallel to the wind tunnel side walls is given. The velocity distribution near the plane of symmetry displays a structure similar to that found in Seddon's data[24], however, for different test conditions. Therefore quantitative comparisons can not be made except to point out some of the salient features of the entire flow field. A reverse flow is observed beneath the lambda shock wave system. The separated flow region begins about three boundary-layer thickness upstream of the normal shock and terminates at five boundary layer thickness downstream. The length of the separated domain is similar to the measurement of Seddon[24] and the numerical simulation by Shea[29].

The entire flow field structure is presented in Figure 10 in terms of density contours at various streamwise locations. The shear layer over the corner region, the strong inviscid-viscous domain, and the subsequent readjustment of the flow field are easily detectable. A clear indication of substantial growth of the shear layer over the wind tunnel wall is also obvious.

Conclusions

A three dimensional time dependent Navier-Stokes code using MacCormack's explicit scheme has been vectorized for the CRAY-1 computer. The vectorized code on CRAY-1 computer achieved a speed of 128 times that of the original scalar code processed by a CYBER 74 computer. The vectorized code

outperforms the scalar code on the CRAY-1 computer by a factor of 8.13.

The numerical simulation for a turbulent, transonic, normal shock-wave boundary-layer interaction in a wind tunnel has been successfully performed using a total 139,400 grid points. The numerical result indicates sufficient resolution for engineering purposes. Additional increase in speed by up to an order of magnitude through algorithm requirement also seems attainable.

Acknowledgement

The authors wish to acknowledge the assistance of Professor D. A. Calahan and Messrs. W. Ames, P. Goshgarian, and E. Sesek of the University of Michigan in preparation of the CRAY-1 program. This work was performed in part under the auspices of Grant AF AFOSR 75-2812 while Mr. Buning was at the University of Michigan. The authors also wish to express their appreciation to CRAY Research, Inc. and Lawrence Livermore Laboratory for the use of their computer facility.

References

1. Chapman, D. R., Dryden Lectureship in Research Computational Aerodynamics Development and Outlook, AIAA Paper 79-0129, January 1979.

2. Peyret, R. and Viviand, H. "Computation of Viscous Compressible Flows Based on the Navier-Stokes Equations," AGARDograph, No. 212, September 1975.

3. Knight, D. D., "Numerical Simulation of Realistic High-Speed Inlets Using the Navier-Stokes Equations," AIAA J., Vol. 16, June 1978.

4. Levy, L. L., "Experimental and Computational Steady and Unsteady Transonic Flow About a Thick Airfoil," AIAA J., Vol. 16, June 1978.

5. Mikhail, A. G., Hankey, W. L. and Shang, J. S., "Computation of a Supersonic Flow Past An Axisymmetric Nozzle Boattail with Jet Exhaust," AIAA Paper 78-993, July 1978.

6. Hung, C. M. and R. W. MacCormack, "Numerical Solution of Three-Dimensional Shockwave and Turbulent Boundary-Layer Interaction," AIAA J., Vol. 16, No. 10, Oct. 1978.

7. Shang, J. S., Hankey, W. L. and Petty, J. S., "Numerical Solution of Supersonic Interacting Turbulent Flow Along a Corner," AIAA Paper 78-1210, July 1978.

8. Pulliam, T. H. and Lomax, H., "Simulation of Three-Dimensional Compressible Viscous Flow on the Illiac IV Computer," AIAA Paper 79-0206, January 1979.

9. "Future Computer Requirements for Computational Aerodynamics," A Workshop held at NASA Ames Research Center, Oct 4-6, 1977, NASA Conference Proceeding 2032.

10. Lin, T. C. and Rubin, S. G., "A Numerical Model for Supersonic Viscous Flow Over a Slender Reentry Vehicle," AIAA Paper 79-0205, January 1979.

11. Schiff, L. B. and Steger, J. L., "Numerical Simulation of Steady Supersonic Viscous Flow," AIAA Paper 79-0130, January 1979.

12. Briley, W. R. and McDonald, H., "Solutions of the Three-Dimensional Compressible Navier-Stokes Equations by a Implicit Technique," Lecture Notes in Physics, Vol. 35, Springer-Verlag, 1975.

13. Warming, R. F. and Beam, R. M., "On the Construction and Application of Implicit Factored Schemes for Conservation Laws," SIAM-AMS Proceedings, Vol. 11, 1978.

14. MacCormack, R. W., "A Rapid Solver for Hyperbolic Systems of Equations," Proceeding of the Fifth International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, 1976.

15. Li, C. P., "A Mixed Explicit-Implicit Splitting Method for the Compressible Navier-Stokes Equations," Proceeding of the Fifth International Conference on Numerical Methods in Fluid Dynamics, Springer-Verlag, 1976.

16. Shang, J. S., "Implicit-Explicit Method for Solving the Navier-Stokes Equations," AIAA J., Vol. 16, May 1978.

17. Buning, P. G., "Preliminary Report on the Evaluation of the CRAY-1 as a Numerical Aerodynamic Simulation Process," Presented at AIAA 3rd Computational Fluid Dynamics Conference, Open Forum, June 1977.

18. Smith, R. E. and Pitts, J. I., "The Solution of the Three-Dimensional Compressible Navier-Stokes Equations on a Vector Computer," Third IMACS International Symposium on Computer Methods for Partial Differential Equations," June 1979, Lehigh University, PA, and Private Communication.

19. Keller, J. D. and Jameson, A., "Preliminary Study of the Use of the STAR-100 Computer for Transonic Flow Calculations," NASA TM 74086, November 1977.

20. Redhed, D. D., Chen, A. W. and Hotovy, S. G., "New Approach to the 3-D Transonic Flow Analysis Using the STAR-100 Computer," AIAA J., Vol. 17, January 1979.

21. MacCormack, R. W., "Numerical Solutions of the Interactions of a Shock Wave with a Laminar Boundary-Layer," Lecture Notes in Physics, Vol. 8, Springer-Verlag, 1971.

22. Calahan, D. A., "Performance of Linear Algebra Codes on the CRAY-1," Proceedings SPE Symposium on Reservior Simulation, Denver, CO, 1979.

23. Abbiss, J. B., East, L. F., Nash, C. R., Parker, P., Pike, E. R. and Swayer, W. G., "A Study of the Interaction of a Normal Shockwave and a Turbulent Boundary Layer Using a Laser Anemometer," Royal Aircraft Establishment, England, TR 75141, February 1976.

24. Seddon, J., "The Flow Produced by Interaction of a Turbulent Boundary-Layer with a Normal Shock Wave of Strength Sufficient to Cause Separation," Royal Aircraft Establishment, England, Rand M 3502, March 1960.

25. Gessner, F. B. and Po, J. K., "A Reynolds Stress Model for Turbulent Corner Flows Part I; Comparisons Between Theory and Experiment," J. of Fluid Engineering, Trans ASME, Vol. 98, Series 1, No. 2, June 1976.

26. CRAY-1 Computer System, Hardware Reference Manual 2240004, CRAY Research, Inc., MN.

27. Higbie, L., "Speeding up Fortran (CFT) Programs on the CRAY-1," Pub. No. 2240207, CRAY Research, Inc., MN.

28. MacCormack, R. W. and Baldwin, B. S., "A Numerical Method for Solving the Navier-Stokes Equations with Application to Shock-Boundary Layer Interactions," AIAA Paper 75-1, January 1975.

29. Shea, J. R., "A Numerical Study of Transonic Normal Shock-Turbulent Boundary Layer Interactions", AIAA Paper 78-1170, July 1978, and Private Communication.

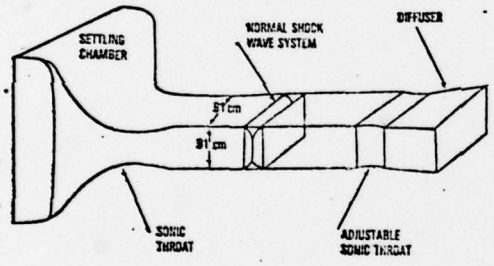

Figure 1. Flow Field Schematic

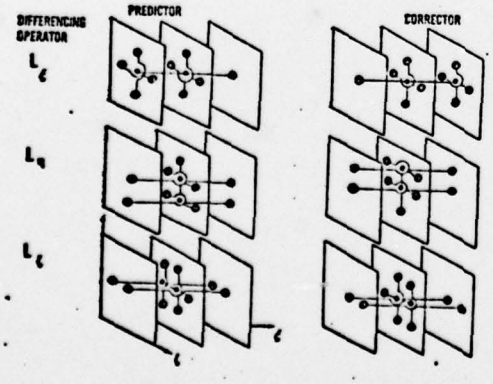

Figure 2. Grid Points Involved in the Time Step Sweep

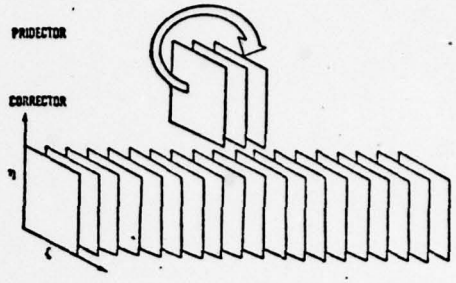

Figure 3. Data Storage and Data Flow Diagram

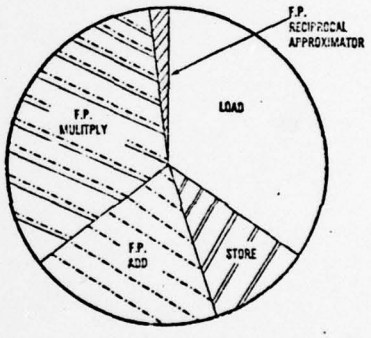

Figure 4. Vector Operation Counts in Percentage

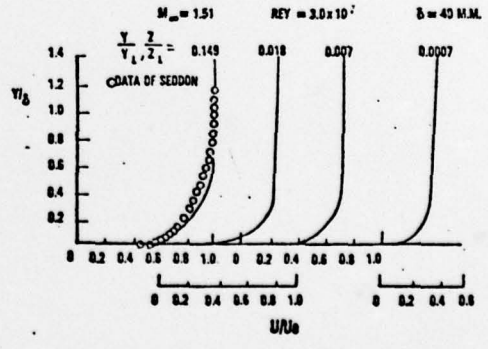

Figure 5. Velocity Profiles Along the Tunnel Wall

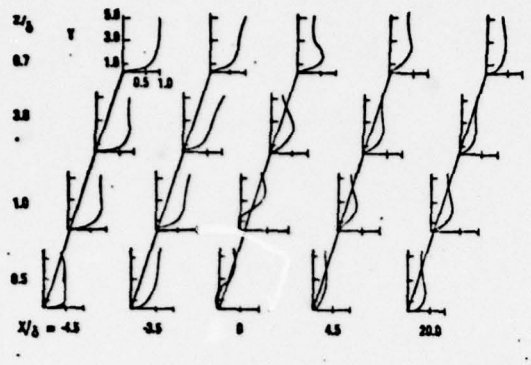

Figure 6. Comparison of the Flow Field Velocity in the Interaction Region

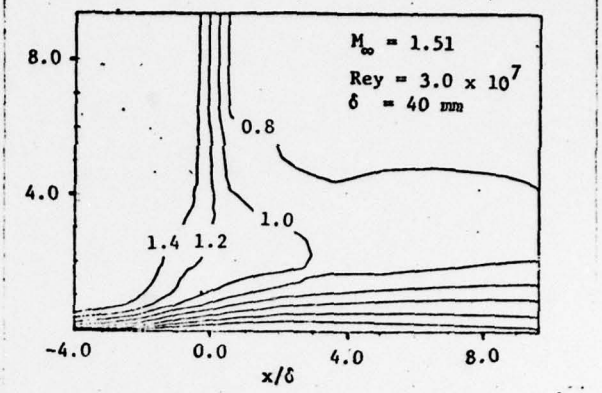

Figure 8. Computed Mach Number Contour in the Plane of Symmetry

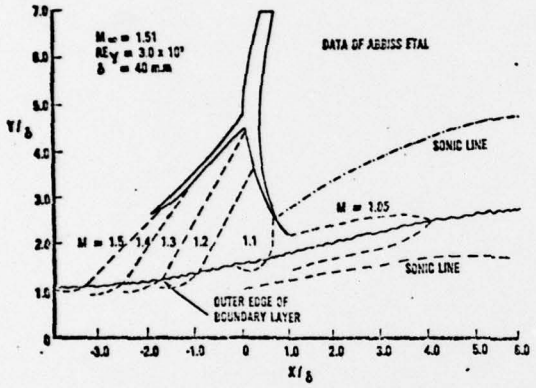

Figure 7. Experimentally Measured Flow Field Structure in the Plane of Symmetry

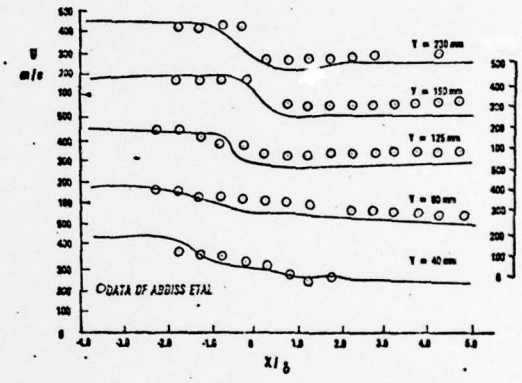

Figure 9. Computed Velocity Field in the Interaction Region

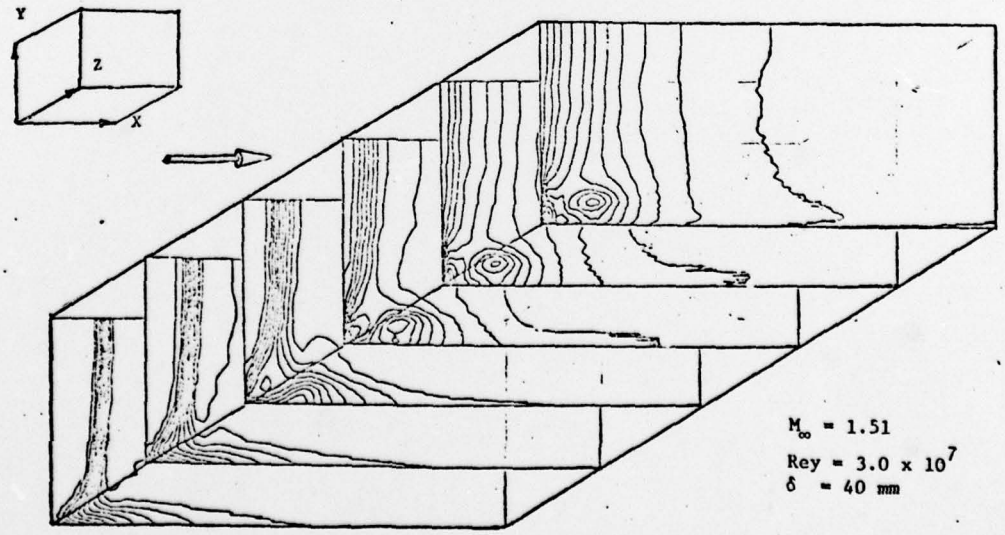

FIGURE 10. Perspective View of Density Contours

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR-79-0790
2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle): THE PERFORMANCE OF A VECTORIZED 3-D NAVIER-STOKES CODE ON THE CRAY-1 COMPUTER.

5. TYPE OF REPORT & PERIOD COVERED: Interim rept.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s): J.S. Shang, W.L. Hankey, P.G. Buning, M.C. Wirth

8. CONTRACT OR GRANT NUMBER(s): AFOSR-75-2812

9. PERFORMING ORGANIZATION NAME AND ADDRESS
University of Michigan
Dept. of Elec. & Computer Engineering
Ann Arbor, Michigan 48109

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 2304/A3

11. CONTROLLING OFFICE NAME AND ADDRESS
Air Force Office of Scientific Research/NM
Bolling AFB, Washington, D.C. 20332

12. REPORT DATE: July 1979

13. NUMBER OF PAGES: 9

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 11p.

15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

Accession For
NTIS GRA&I
DDC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or special
A

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Vector processors, Parallel computation, Fluid dynamics, Aerospace simulation.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A three-dimensional, time dependent Navier-Stokes code using MacCormack's explicit scheme has been vectorized for the CRAY-1 computer. Computations were performed for a turbulent, transonic, normal shock wave boundary layer interaction in a wind tunnel diffuser. The vectorized three-dimensional Navier-Stokes code on the CRAY-1 computer achieved a speed of 128 times that of the original scalar code processed by a CYBER 74 computer. The vectorized version of the code outperforms the scalar code on the CRAY computer by a factor of 8.13. A comparison between the experimental data and the numerical simulation is also made.

DD FORM 1 JAN 73 1473 407 400 UNCLASSIFIED LB